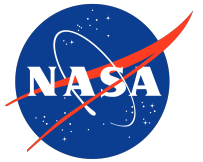


V&V for Autonomy

Dr. Guillaume Brat

NASA Ames Research Center

Intelligent Systems Division

Robust Software Engineering

Introduction

- CAST: Core Avionics and Software Technologies
  - cFS/cFE based FSW
  - Model-based development
  - Swarm simulation
  - Small spacecraft swarm
  - FSW Monitoring & Review
- ESMD: LCROSS, LADEE
- HEOMD: BioSentinel
- SMD: RP, VIPER
- STMD: Starling, DSA
- ARC: CASAS
- ASSET: Automated Software & System Engineering Technologies
  - Formal methods
    - Requirement Elicitation/Analysis
    - Design Model Analysis
    - Static Code Analysis
    - Run-Time Analysis
  - Adaptive stress testing
  - Assurance Cases
  - Machine learning V&V
- ARMD
  - AOSP: System-Wide Safety
    - Aviation software assurance
    - Autonomous systems assurance
  - Low Boom Flight Demonstrator
- Others
  - Gateway contractors V&V
  - BlueOrigin: ROS 2 V&V
  - DoD Autonomous Systems T&E
- Traditional Aviation
  - US Airframers, OEMs, Airlines
  - FAA, NTSB
  - AFRL
- Emerging Markets
  - Urban Air Mobility
  - Plant inspection, disaster response, precision agriculture, USGS

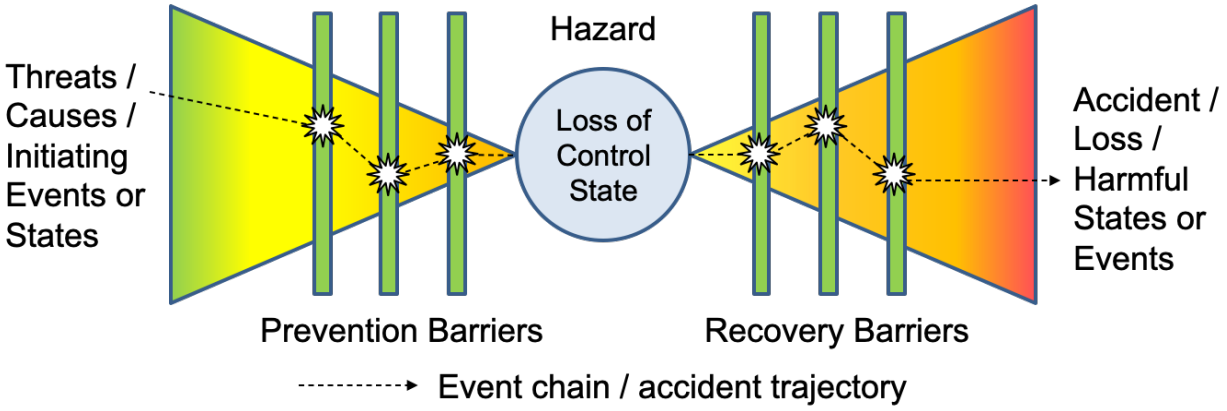
Focus elements

- Our strategy for the V&V of autonomy is focusing on the following elements:
  - Explaining why a system is safe, and remains safe in operations, through dynamic safety cases
  - Identifying more hazards, especially when a system is complex or adapting to changing conditions
  - Formalizing requirements to facilitate safety analysis and traceability
  - Computing “provenly” safe bounds of operations
  - Addressing verification of machine learning components

Dynamic Safety Cases

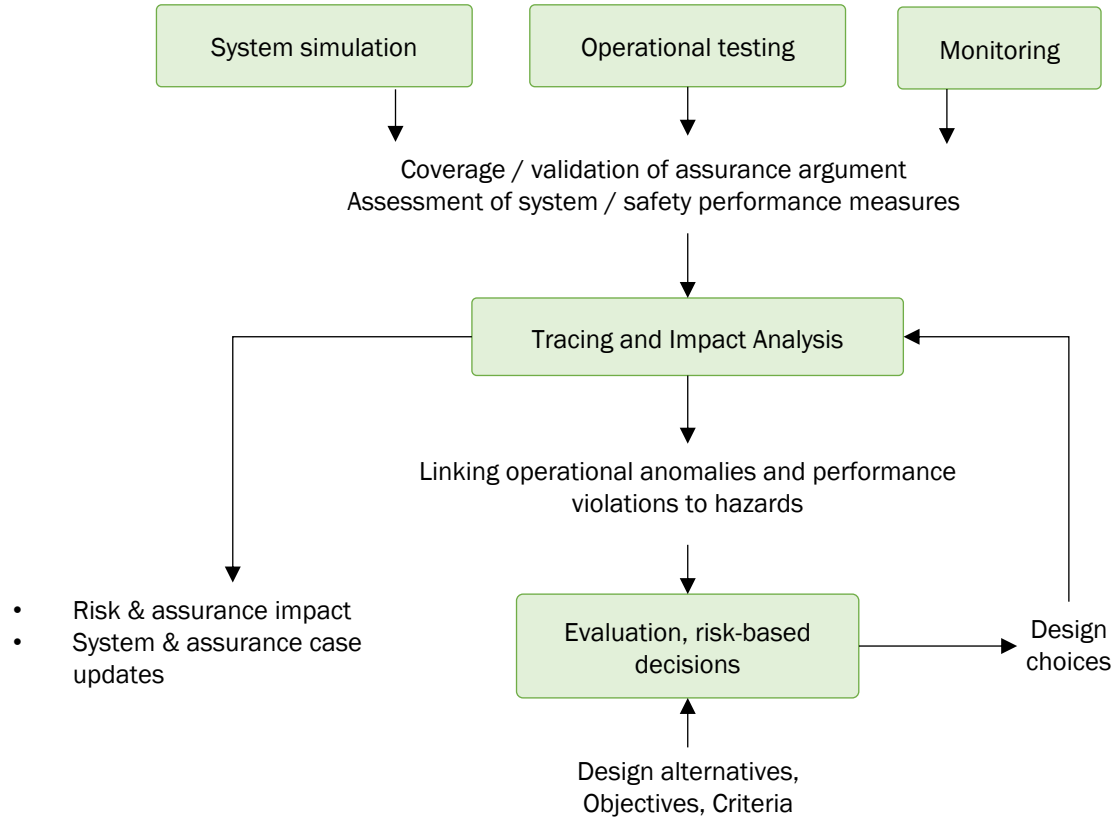
> 'A safety case is a structured argument, supported by a body of evidence, that provides a compelling, comprehensible and valid case that a system is safe for a given application in a given operating environment'
> - *NASA System Safety Handbook ver. 1 (2014)*

- We have developed a tool called AdvoCATE for creating and visualizing safety cases
- AdvoCATE Barrier models
  - Risk scenarios showing chain of events leading to accidents, loss, or harm
  - Represented using Bow Tie Diagrams
  - Barrier = Risk reduction mitigation
    - Prevention or recovery function
    - Reduction of event probability and severity



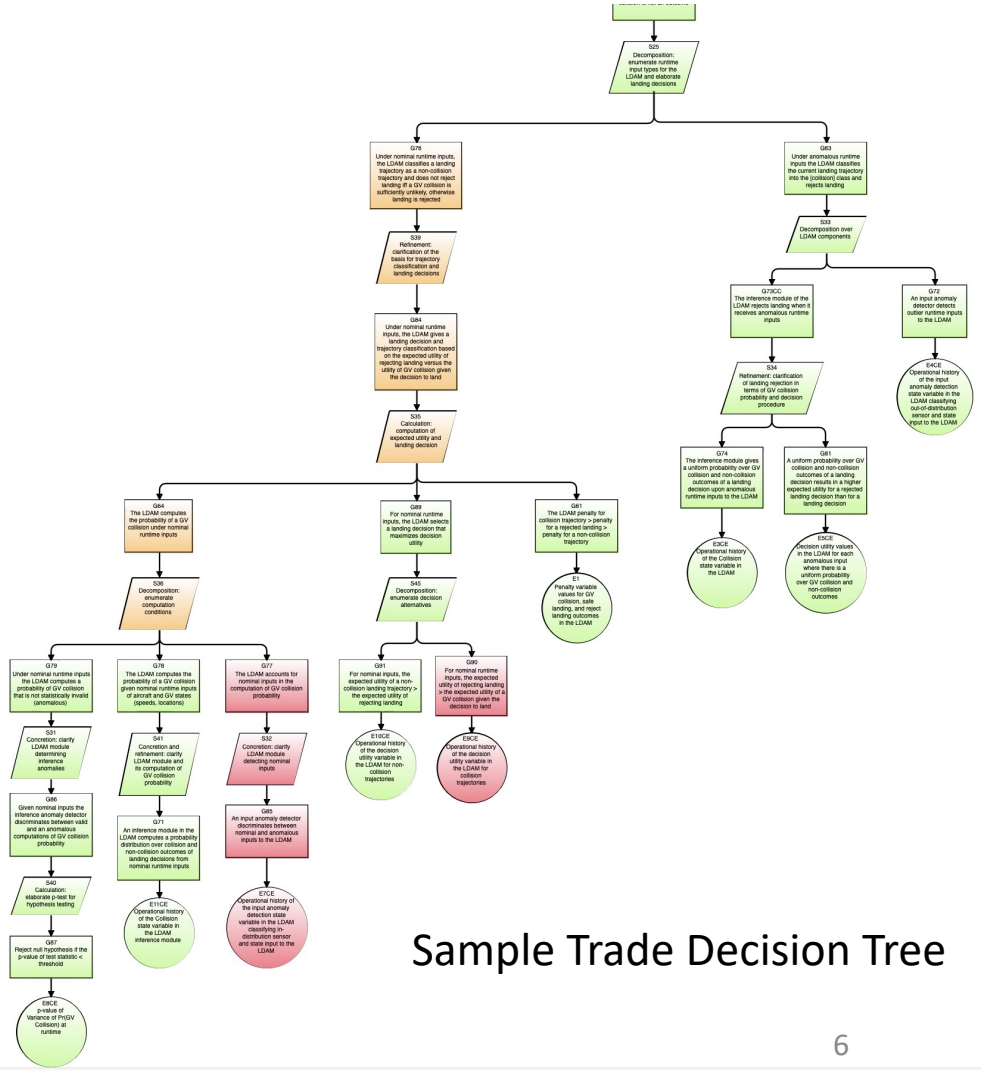
Dynamic Safety Cases

- Extend AdvoCATE into a dynamic dashboard to reason about safety during operations and support risk-based decision making
- Integration of risk-based design, development, and assurance in AdvoCATE
- Demonstrated in March 2021 on a centerline tracking example implemented with DNNs



Dynamic Safety Cases

- Trade trees record decision alternatives (e.g., design decisions) to be presented with
  - impact of each alternative on baseline risk levels, as well as on existing assurance artifacts,
  - decisions about alternatives to be made and recorded, and
  - justifications of those decisions to be provided as assurance arguments.
- Integration of decision-making, assurance impact analysis, and (safety) assurance case development within a common framework.



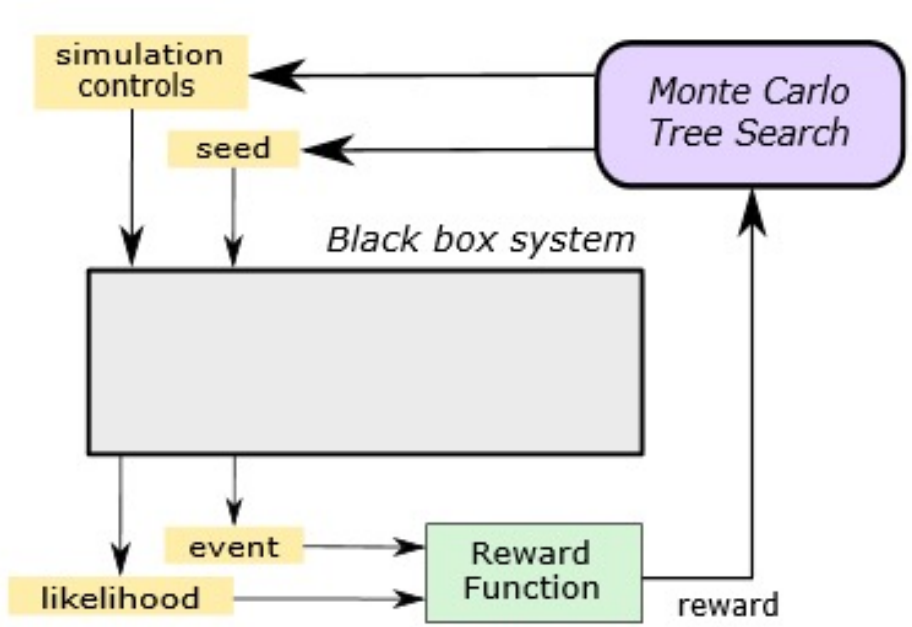
Sample Trade Decision Tree

Identifying More Hazards

- Identifying more hazards, especially when a system is complex or adapting to changing conditions
  - Adaptive Stress (AdaStress) testing
  - Functional and fault modelling with Fmdtools
  - Use of NLP for mining historical data and identify hazards that are often overlooked

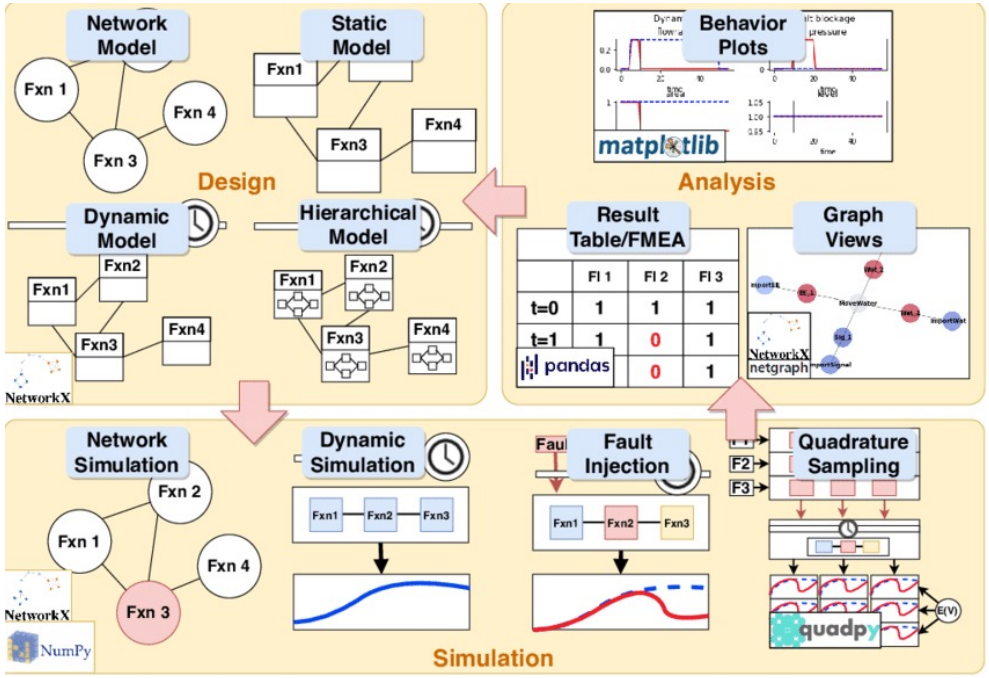
Identifying More Hazards: AdaStress

- AdaStress (Adaptive Stress testing) is a software package for an accelerated simulation-based stress testing method for finding the most likely path to a failure event
- It has been successfully demonstrated to industry
  - It has been adopted by GE Aviation Systems and is being incorporated in their continuum testing environments.

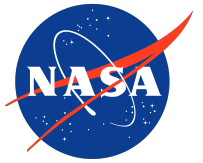


Identifying More Hazards: Fmdtools

- Fmdtools (Fault Model Design *tools*) is a design and analysis environment, which enables a designer to
  - represent the system in the early design process,
  - simulate the effects of faults, and
  - quantify corresponding resilience metrics.

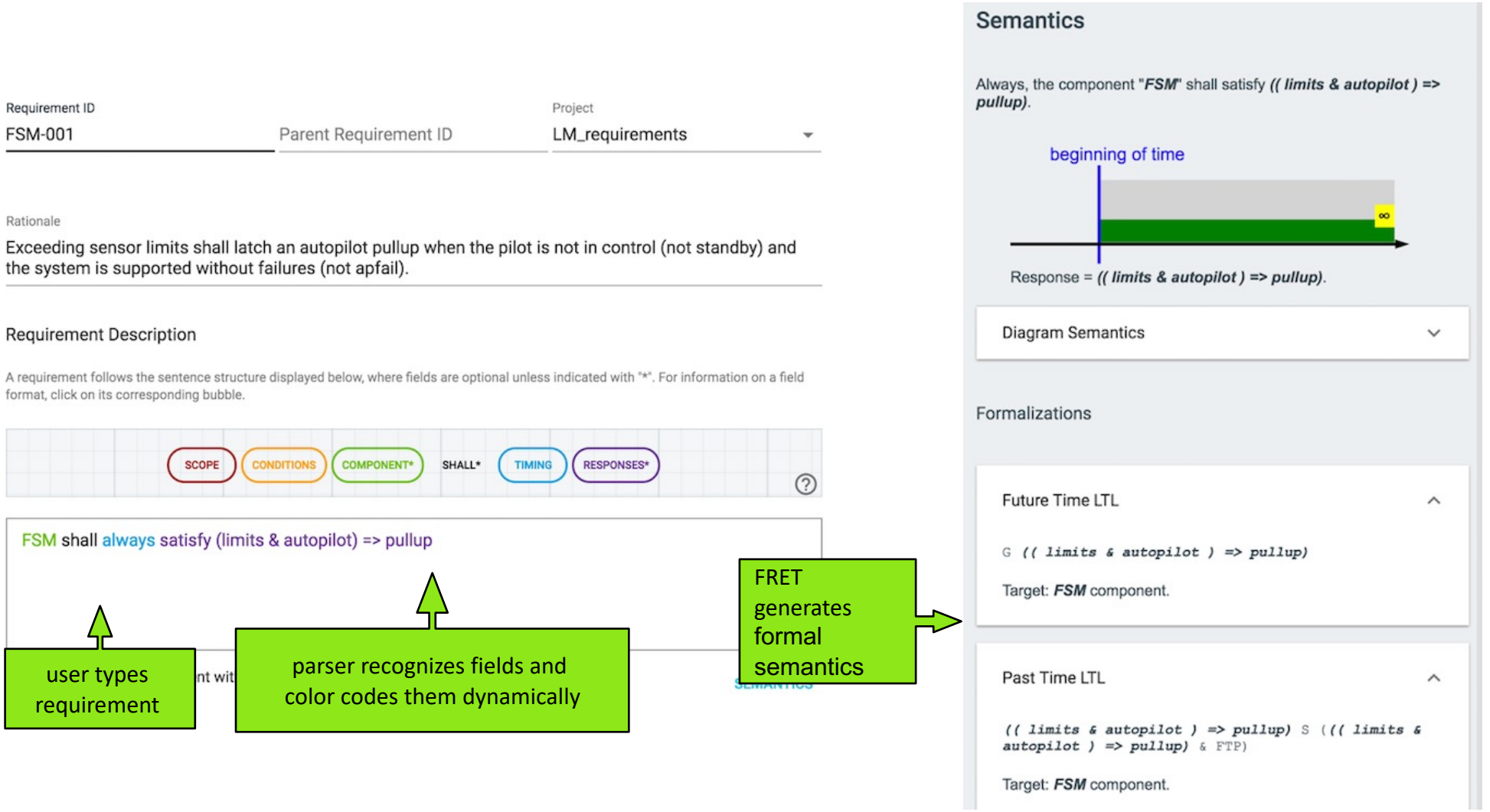


Formalizing Requirements



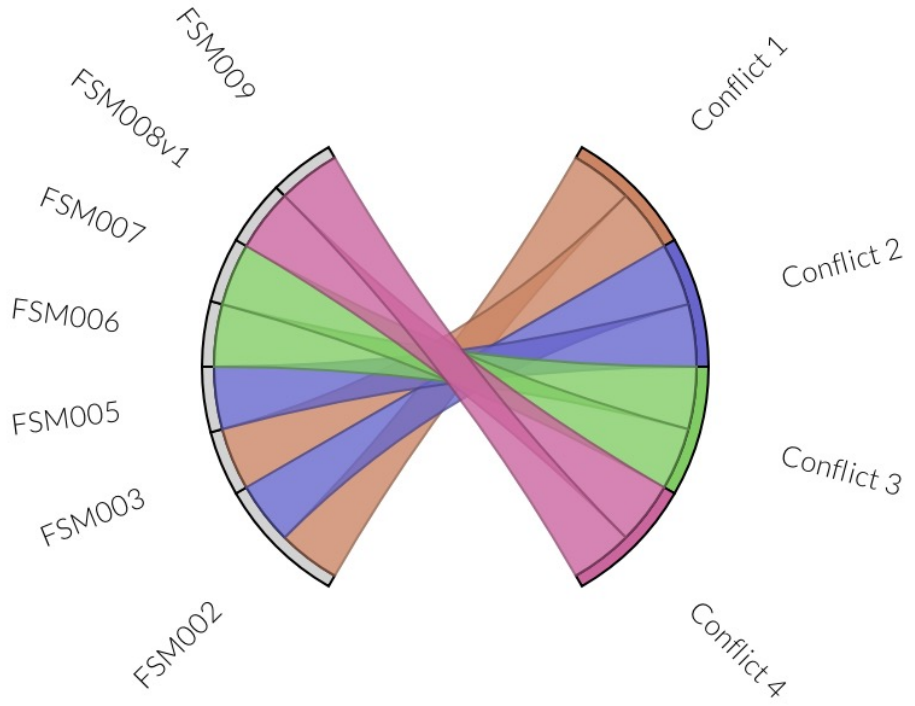
- We have developed a tool called FRET that allows a user to specify requirements in a quasi-natural language in order to formalize them into various logics
- This helps support formal analysis in later stages of the development process
  - E.g., for formalizing requirements that we want to prove on Simulink models using the CoCoSim tool.
- It also allows us to perform formal analysis on requirements
  - E.g., detecting inconsistencies between requirements

Formalizing Requirements



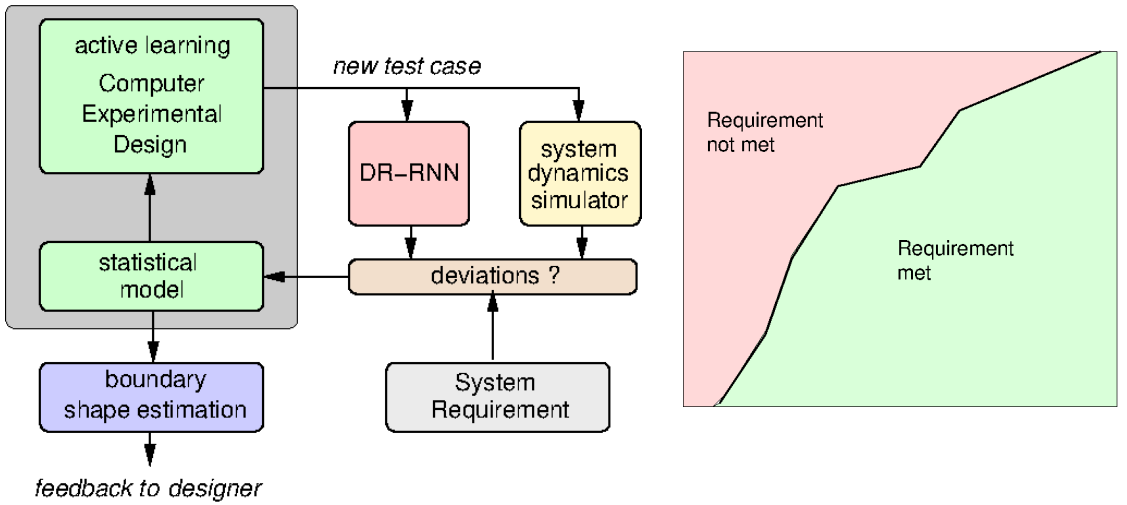
Formalizing Requirements

- Checking realizability of requirements, which aims at determining whether an implementation exists, always complying with a set of requirements, regardless of the stimuli provided by the system's environment.



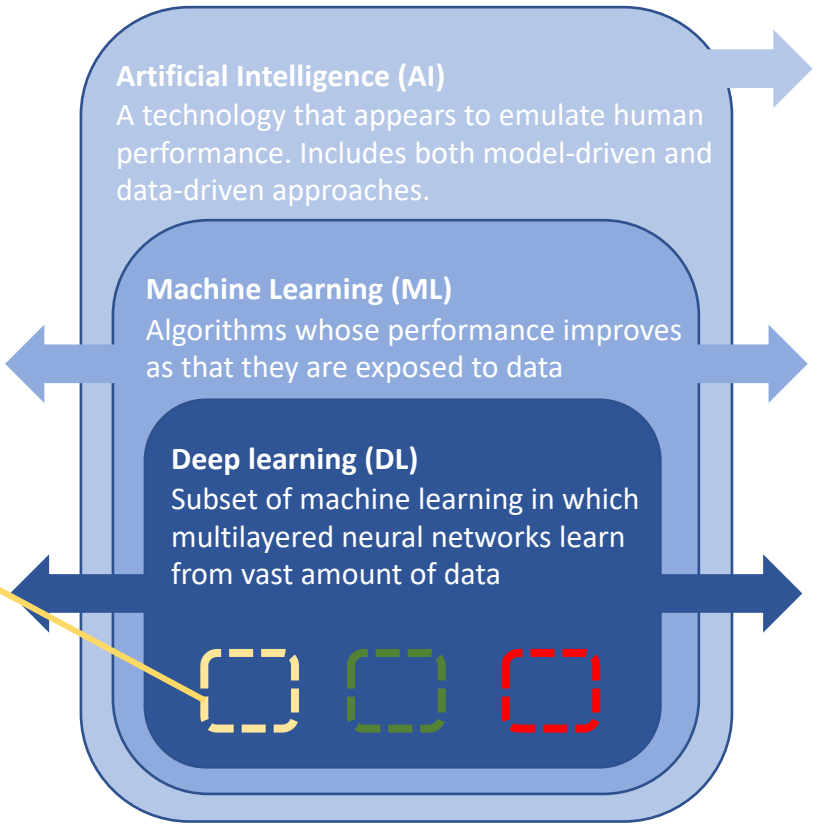
Computing Safe Bounds of Operations

- Perturbations in the operational phase due to fluctuations in the data input and prediction output or in the model itself.
  - Use of SysAI (new branch of MARGInS) to evaluate the Taxinet example with noisy data.
  - Perturbations in the design phase could also be found using SysAI.



V&V of Machine Learning

- We focus mostly on off-line-trained, supervised DNNs
- Some of our results apply only to ReLU DNNs



Field of application

E.g., Expert System

E.g., Classification (Clustering)

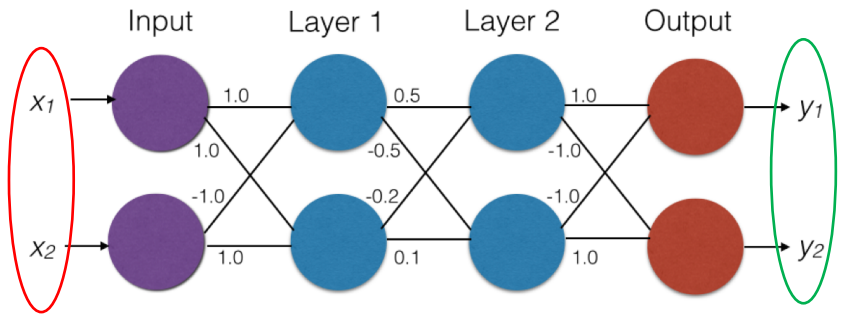
E.g., Computer vision (CNNs) or Natural Language Processing (RNNs)

Type of ML Supervised Unsupervised Reinforcement

V&V of Machine Learning

- Prophecy Key Idea:
  - Infer “likely” properties, aka contracts, of a NN
  - Prove them using a decision procedure
- What is a contract? σ => P
  - σ is a precondition (“safe region”)
  - P is a postcondition; desired output behavior (e.g. some prediction)

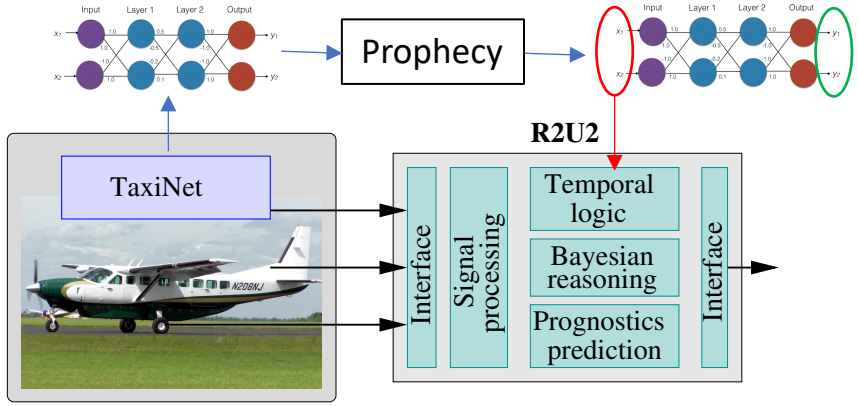
If σ holds for



satisfies P

V&V Techniques Working together

- Use of runtime monitoring (R2U2 tool) to monitor inputs based on pre-conditions computed by Prophecy
  - Or by MARGInS or SysAI



- R2U2 is a run-time monitoring and V&V tool that combines Metric Temporal Logic observers, Bayesian Network reasoners, and model-based prognostics.
- FRET can also be used to generate formal monitors for R2U2 based on safety requirements.

Conclusions

- RSE is developing tools to make the software and system engineering processes more efficient.
- Most of the funding has been coming from ARMD, hence a focus on aviation systems
- Since we believe they make a difference, we are also injecting these tools in our small Sat missions
- We have created an internal mission called Troupe1, which we use to
  - Train new staff for missions
  - Evaluate the use of new tools and processes

Image courtesy of NASA Ames Research Center

RSE Software Engineering Tools

Tools	Description	Availability	Technical POC	POC Email
FRET	Requirement elicitation and analysis	Open Source	Dimitra Giannakopoulou	dimitra.giannakopoulou@nasa.gov
CoCoSim	Simulink model analyzer	Open Source	Khanh Trinh	khanh.v.trinh@nasa.gov
IKOS	Static code analysis for C/C++	Open Source	Guillaume Brat	guillaume.p.brat@nasa.gov
AdvoCATE	Assurance case automation toolset	Open Source	Ewen Denney	ewen.w.denney@nasa.gov
MARGInS	ML/statistical libraries for system testing	Usage Agreement	Carlos Paradis	carlos.v.paradis@nasa.gov
SysAI	ML/statistical libraries for system testing	Not available yet	Yuning He	yuning.he@nasa.gov

RSE Software Engineering Tools

Tools	Description	Availability	Technical POC	POC Email
AdaStress	Adaptive stress testing	Open Source	Ritchie Lee	ritchie.lee@nasa.gov
RACE	Runtime for Airspace Concept Evaluation	Open Source	Peter Mehlitz	peter.c.mehlitz@nasa.gov
MESA	Run-time analysis of live data streams	Open Source	Nastaran Shafiei	nastaran.shafiei@nasa.gov
Prophecy	Formal analysis of Neural Networks	Not available yet	Corina Pasareanu	corina.s.pasareanu@nasa.gov
R2U2	Vehicle-level run-time analysis	Usage Agreement	Johann Schumann	johann.m.schumann@nasa.gov
Drishti	NLP for requirement traceability	Not available yet	Nija Shi	nija.shi@nasa.gov

Other Topics From Ames Research Center

TECHNICAL SESSIONS

- Mission Operations and Autonomy: *Design and Testing of Autonomous Distributed Space Systems*
- Science/Mission Payloads: *The Pandora SmallSat: Multiwavelength Characterization of Exoplanets and their Host Stars*
- Constellation Missions: *Design and Validation of an Autonomous Mission Manager towards Coordinated Multi-Spacecraft Missions*

NASA SHORT TALKS

- *Framework for Autonomous Planning of Distributed Space Systems*
- *How to Partner with NASA and Use Patented Technologies*
- *NASA Art in Space: A Rich History and The PACE-1 Flight Mission*
- *NASA Flight Opportunities: Competitive Access to Suborbital Flight Testing*
- *NASA SSTP's SmallSat Technology Partnerships for Universities*
- *Novel Communication Experiments in the Nano-Orbital Workshop (NOW) Series, Closing the Link to Geostationary Orbit with Automated Doppler Correction*